Efficient Functional Decomposition Algorithm Based on Indexed Partition Calculus

Mariusz Rawski, Paweł Tomaszewicz, and Piotr Szotkowski

1 Introduction

Functional decomposition is one of the best logic synthesis method targeting FPGAs. The most popular decomposition algorithms may be roughly divided into ones using *binary decision diagrams* (BDD) for Boolean function representation and ones using *cube* representation of the decomposed function. The advantage of the former is an efficient representation of functions with a large number of input variables. However, these algorithms face the problem of representing multiple-output and not fully specified functions [8, 9]. The latter algorithms allow simple representation of multiple-output and not fully specified functions. Most algorithms based on this representation use the blanket calculus for function manipulation [1]. The blanket calculus allows construction of very efficient decomposition algorithms [7, 10]. However the computational complexity of blanket manipulations makes it impossible to efficiently apply these algorithms to large Boolean functions.

In [5] a concept of *indexed partition* has been presented, allowing to manipulate Boolean functions described by cubes in similar way as blanket calculus but requiring much less computational and memory complexity. It combines the advantages of both blanket calculus and information relationships and measures [2].

2 Indexed Partition Calculus

Blankets, as well as information sets, are the way of expressing the compatibility relation COM on Boolean function's cubes. This also can be done using the concept of compatibility or *incompatibility graph*.

M. Rawski (✉) • P. Tomaszewicz • P. Szotkowski
Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
e-mail: rawski@tele.pw.edu.pl

Operations on blankets have direct analogue in operations on incompatibility graphs. The concept of *indexed partitions* is used to model such operations on incompatibility graphs.

Definition 1. An *indexed partition is a set of ordered dichotomies* $\{B_i, P_i\}$ *on set S, such that* B_i *are disjoint subsets of S* and

$$\bigcup_i B_i = S. \tag{1}$$

In [1] a theorem based on the concept of cubes has been proposed which describes the existence of serial decomposition. This theorem can be re-expressed using the concept of indexed partitions.

Theorem 1. *Let* δ_V, δ_U, *and* δ_F *be indexed partitions induced on the function's F input cubes by the input subsets V and U, and outputs of F, respectively.*

If there exists an indexed partition δ_G *on the set of function F's input cubes such that* $\delta_V \leq \delta_G$, *and* $\delta_U \cdot \delta_G \leq \delta_F$, *then F has a serial decomposition with respect to (U,V).*

In practice, even large Boolean functions of many input variables are often specified by relatively small number of cubes. Since the size of indexed partition depends on the number of cubes, the test from Theorem 1 can be generally much computationally simpler than the test based on blankets [5].

3 Functional Decomposition Algorithm

The algorithm presented in this paper is based on the method of decomposition of function sets presented in [4]. However the blanket calculus has been substituted by indexed partition calculus.

The cube representation can easily describe a multi-output, not fully specified Boolean function using the

H. Selvaraj et al. (eds.), *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*, Advances in Intelligent Systems and Computing 1089,
DOI 10.1007/978-3-319-08422-0_117,

espresso format. Unfortunately, it can be very inefficient when single outputs of the multi-output function depend on different input variables. The method proposed in [4] allows decomposing single-output Boolean function sets instead of one multi-output function.

The logic multi-output circuit of n input variables X can be represented as a set of Boolean functions $f_1(X_1), \ldots, f_m(X_m)$, where X_i is a sub-set of X. In the general case, each single function may depend on different subset of input variables and can be described by a different set of cubes. The application of serial functional decomposition based on blanket calculus is impossible in this case. The proposed modified serial functional decomposition algorithm allows decomposing function sets described by separate truth tables, even if each single function depends on different subset of input variables and truth tables consist of different sets of cubes.

When decomposing a set of Boolean functions $f_1, \ldots, f_m$ it is necessary to find such sub-function G that satisfies the decomposition condition for each function: $f_i = h_i(U_i, G(V))$.

The decomposition algorithm consists of the following steps:

1. input variable partitioning an appropriate input support V has to be selected for function G; in case functions $f_1, \ldots, f_m$ are specified by different truth tables, the V set has to be selected in such a way that it is a subset of input variables of each decomposed function,
2. the computation of δ_{V_i}, δ_{U_i} and δ_{f_i} for each function f_i - since truth tables of each function may consist different cubes, the indexed partitions δ_{V_i}, δ_{U_i} and δ_{f_i} for each function may be different,
3. the construction of δ_{G_i} in functional decomposition algorithm the construction of this indexed partitions corresponds to the construction of the multi-valued function G, since we want to find the decomposition with the same function G for all functions f_i, such δ_{G_i} should be created that satisfy the decomposition condition for all these functions and correspond to the same function G at the same time.

This decomposition method allows decomposing a set of Boolean functions described by separate truth tables. The decomposed function may depend on different input variables and can be described by different set of cubes. The application of the proposed decomposition algorithm allows for efficient creation of sub-functions common to all decomposed functions. This allows the designer to apply a much wider range of synthesis strategies.

Table 1 Results of Synthesis for 6-input LUTs

			Imfs+Lutpack		CombDec	
example	inputs	outputs	LUTs	levels	LUTs	levels
alu4	14	8	453	5	116	8
apex4	9	19	732	5	172	3
ex1010	10	10	1059	5	159	4
ex5	8	63	108	3	114	2
misex3	14	14	446	5	116	6
pdc	16	40	171	5	127	4
spla	16	46	263	4	153	5
$\sum$			3232	32	957	32

4 Results

For the purpose of evaluation of the efficiency of the proposed decomposition algorithm, a software tool *CombDec* has been created. The tool implements multilevel logic synthesis based on the concept of decomposition of function sets. To solve the input variable partitioning problem, the heuristic method presented in [6] has been applied.

The tool has been used to decompose selected benchmark examples into 6-input LUT networks. The resulting LUT networks were all verified using the combinational equivalence checker implemented in *ABC*.

Table 1 presents the comparison of the results obtained by the presented approach and the results reported in [3], obtained by technology mapping and resynthesis procedures (Imfs+Lutpack) implemented in the *ABC* tool. For both tools the number of 6-input LUTs in the resulting networks and the delay calculated as the depth of the networks have been provided.

It can be noticed that the method proposed in this paper produces networks of much greater area quality (calculated as the number of 6-LUTs) than those generated by the *ABC* procedures. The delay of networks obtained by both methods is similar.

5 Conclusions

There are many methods for decomposition of multi-output Boolean functions. These methods mostly use cube or BDD-based representation of the decomposed function. However, the common characteristics of these methods is the requirement that the decomposed multi-output Boolean function is represented by a single truth table in the case of cube representation or a single BDD.

The proposed decomposition method allows decomposing a set of Boolean functions described by separate truth tables. Application of indexed partition allows performing multi-level logic synthesis for multi-output, not fully specified Boolean functions of large number of input variables. The obtained results are of high quality in terms of the area calculated as the number of LUTs, as well as the delay of resulting network.

References

1. J.A. Brzozowski and T. Łuba. Decomposition of boolean functions specified by cubes. *Journal of Multiple-Valued Logic and Soft Computing*, 9:377–417, 2003.
2. L. Jóźwiak. Information relationships and measures: an analysis apparatus for efficient information system synthesis. In *EUROMICRO'97. New Frontiers of Information Technology., Proceedings of the 23rd EUROMICRO Conference*, pages 13–23, September 1997.
3. A. Mishchenko, R. Brayton, and S. Chatterjee. Boolean factoring and decomposition of logic networks. In *Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on*, pages 38–44, November 2008.
4. M. Rawski. Decomposition of boolean function sets. *Electronics and Telecommunications Quarterly*, 53(3), 2007.
5. M. Rawski. Application of indexed partition calculus in logic synthesis of boolean functions for FPGAs. *International Journal of Electronics and Telecommunications*, 57(2), 2011.
6. M. Rawski. Heuristic algorithm of bound set selection in functional decomposition for heterogeneous FPGAs. In *Systems Engineering (ICSEng), 2011 21st International Conference on*, pages 465–466, 2011.
7. M. Rawski, T. Łuba, Z. Jachna, and P. Tomaszewicz. The influence of functional decomposition on modern digital design process. *Design of Embedded Control Systems*, pages 193–203, 2005.
8. T. Sasao, Y. Iguchi, and M. Matsuura. Comparison of decision diagrams for multiple-output logic functions. In *International Workshop on Logic and Synthesis*, pages 4–7, 2002.
9. T. Sasao and M. Matsuura. BDD representation for incompletely specified multiple-output logic functions and its applications to functional decomposition. In *IN PROC. DESIGN AUTOMATION CONF*, pages 373–378, 2005.
10. H. Selvaraj, P. Sapiecha, M. Rawski, and T. Łuba. Functional decomposition - the value and implication for both neural networks and digital designing. *International Journal of Computational Intelligence and Applications*, 6(1):123–138, 2006.